

**Final Report: The Center for Space
Telemetry and Telecommunications
Systems**

by

**S. Horan, P. DeLeon, D. Borah, and R. Lyman
NMSU-ECE-03-003**

NMSU-ECE-03-003

Final Report: The Center for Space Telemetry and Telecommunications Systems

S. Horan, P. DeLeon, D. Borah, and R. Lyman
Manuel Lujan Space Tele-Engineering Program
New Mexico State University
Las Cruces, NM

Prepared for

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD 20771

under Grant NAG5-9323

July 10, 2003



Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 30001, MSC 3-O
Las Cruces, NM 88003-8001

CONTENTS

SUMMARY	1
1 Degree Production	2
2 Technical Papers	2
3 Technical Reports	3
4 Space Protocol Testing	4
5 Autonomous Reconfiguration of Ground station Receivers.....	19
6 Satellite Cluster Communications	31
7 Bandwidth-Efficient Modulation	32

SUMMARY

This report comprises the final technical report for the research grant “Center for Space Telemetry and Telecommunications Systems” sponsored by the National Aeronautics and Space Administration’s Goddard Space Flight Center. The grant activities are broken down into the following technology areas:

- Space Protocol Testing
- Autonomous Reconfiguration of Ground Station Receivers
- Satellite Cluster Communications, and
- Bandwidth Efficient Modulation

The grant activity produced a number of technical reports and papers that were communicated to NASA as they were generated. This final report contains the final summary papers or final technical report conclusions for each of the project areas.

Additionally, the grant supported students who made progress towards their degrees while working on the research.

1 Degree Production

The following students received degrees based on research conducted as part of this grant during the final grant year:

- A. Chakraborti – MSEE; Project: Performance Analysis Of MDP In Simulated Space Channel Environment
- S. Muddasani – MSEE; Project: Assessment of the Multicast Dissemination Protocol for Constrained Satellite Channels
- S. Narina – MSEE; Project: Optimal Point-to-Point Protocol (PPP) Frame Size for Transmission Control Protocol (TCP) in High Error Environments

2 Technical Papers

The following technical papers were presented by the faculty as part of the grant activities for this final grant year:

- D. K. Borah and S. Horan, "Performance Study of Enhanced FQPSK and Constrained Envelope Modulation Techniques," *Proc. International Telemetry Conf.*, ITC'2002, San Diego, USA, Oct. 2002.
- Q. Zhao and D. K. Borah, "Adaptive Iterative CDMA Multiuser Detection in Unknown Multipath Channels," *Proc. IEEE Int. Commun. Conf.*, ICC2002, April 2002, New York.
- S. Horan, L. Alvarez, et al., "The Three Corner Satellite Mission," *Proc. International Symposium Formation Flying: Missions & Technologies*, Toulouse, FR, October 2002.
- S. Horan and L. Alvarez, "Preparing a COTS Radio for Flight - Lessons Learned from the 3 Corner Satellite Project," *Proc. 16th Annual/USU Conference on Small Satellites*, SSC02-X-4, Logan, UT, August 2002.
- S. Horan, et al., "Evaluation of MDP Over Low Bandwidth-Delay Product Links," 2nd Space Internet Workshop, Greenbelt, MD, May 2002.

The following book and book chapter were influenced by the research conducted under this grant:

- S. Horan, "Telemetry Systems," in *The Engineering Handbook* 2nd ed., Boca Raton: CRC Press, in press.
- S. Horan, *Introduction to PCM Telemetry Systems*, 2nd ed., Boca Raton: CRC Press, 2002.

The following journal articles were published based on research from this grant:

- S. Horan, "Non-Tracking Antenna Performance for Inertially Controlled Spacecraft Using TDRSS," *IEEE Trans. on Aerospace & Electronic Systems*, in press.
- S. Horan, "The Potential for Using LEO Telecommunications Constellations to Support Nanosatellite Formation Flying," *International Journal of Satellite Communications*, **20**, 2002, p. 347 - 361.
- S. Horan and R. Wang, "Design of a Space Channel Simulator Using Virtual Instrumentation Software," *IEEE Trans. Instrument and Measurements*, Vol. 51, No. 5, October 2002, p. 912-916.

3 Technical Reports

The following technical reports were submitted as part of the research effort conducted during this final grant year:

- D. K. Borah, "Spectral Re-growth Reduction for CCSDS 8-D 8-PSK TCM," NMSU-ECE-02-006, October 2002.
- A. Chakraborti and S. Horan, "Performance Analysis Of MDP In Simulated Space Channel Environment," NMSU-ECE-02-009, December 2002.
- S. Muddasani and S. Horan, "Assessment of the Multicast Dissemination Protocol for Constrained Satellite Channels," NMSU-ECE-03-002, April 2003.
- S. Horan, "Satellite Cluster Communications," NMSU-ECE-02-004, September 2002.
- S. Narina and S. Horan, "Optimal Point-to-Point Protocol (PPP) Frame Size for Transmission Control Protocol (TCP) in High Error Environments," NMSU-ECE-02-003, May 2002.
- S. Horan, "Protocol Testing Procedures," NMSU-ECE-02-002, May 2002.

4 Space Protocol Testing

The space protocol testing effort moved from testing Consultative Committee for Space Data Systems (CCSDS) Space Communications Protocol Standards (SCPS) into those based on UDP protocols. The testing in this area complements similar tests conducted at GSFC and the results are summarized in the following paper that has been submitted for publication consideration in a journal being organized by Marjory Johnson at NASA Ames Research Center.

Testing MDP in a Simulated Space Channel Environment

Stephen Horan, Anirban Chakraborti, Sandeep Muddasani, and Srinivasulu Narina
Klipsch School of Electrical and Computer Engineering
Box 30001, MSC 3-O
New Mexico State University
Las Cruces, NM 88003¹

Abstract

The performance of the Multicast Dissemination Protocol (MDP) in a simulated satellite channel is investigated. The environment that is modeled is that to be expected in a small satellite communications environment including channel errors and link interruptions. The satellite link is simulated with a channel environment simulator. The performance of MDP is compared with file transfer protocol (*ftp*) acting as a standard basis. In these experiments, we find that MDP does not perform as well as *ftp* in the high quality channel, although the performance difference is small. MDP does perform better than *ftp* in low-quality channels. MDP is also able to resume data transmissions when a link outage occurs. MDP is seen as being a viable option for use in small satellite communications systems.

INTRODUCTION

The use of Internet-type protocols for space communications over bent-pipe links has been well studied and the expected performance well characterized. Much of the research has examined the performance of the Transmission Control Protocol (TCP) in space from observations of simulation and experimental testbed results. See, for example, [1], [2], and [3]. The purpose of these studies was to identify a better TCP variant for use in long-delay networks, such as those found in the satellite environment and to investigate the effect of latency on aggregated network utilization.

One next logical evolution in the process is to consider the satellite to be another node on the Internet. Research on this subject has been conducted at government, contractor, and university facilities for several years now; see, for example, [4], [5], and [6], and references therein. The intent of this mode is not only to use the satellite as part of the infrastructure backbone but to also have the ability to terminate the link within the satellite itself to support space operations. As part of our satellite communications research program, we have been investigating the performance of both connection-oriented and connectionless protocols for satellite data transfer to a ground control point. In our studies, we have been concentrating on the performance of the protocols in a small satellite environment.

The space channel environment poses a number of challenges to providing reliable, end-to-end data communications with a user-specified level of service. Losses due to

¹ This effort was funded under grants NAG5-7520 and NAG5-9323 from the National Aeronautics and Space Administration.

transmission errors, large point-to-point propagation delays, constrained bandwidth, asymmetric link data rates, and intermittent link connectivity all conspire to severely limit the performance of any ground-network-based protocol. These effects are problematic for near-earth satellites and are a major impediment for links with long delays [7]. In particular, TCP-based protocols are known to have problems with confusing channel errors with link congestion and thereby causing a reduction in transmission efficiency when the protocol invokes its rate back-off mechanism. TCP also typically uses some form of slow-start or ramp-up mechanism at the beginning of the transmission as it probes for the link congestion state. Since the typical satellite communications channel with one node originating or terminating at the satellite is an uncongested link, these mechanisms reduce link transmission efficiency as well.

While TCP-based protocols can have the advantage of end-to-end connection management, the inefficiency in link utilization to support the management may not be appropriate in satellite channels where the pass time, and hence the connection life, may be short. One way to improve efficiency is to consider Unconnected Datagram Protocol (UDP) based transmission architectures. The UDP protocols will typically use a best-effort means to deliver the data but the delivery is not guaranteed. UDP will then have the expected advantages of minimal overhead and maximum data flow when compared with connection-oriented protocols. Conversely, the UDP-based protocols will not be as reliable as connection-oriented protocols without additional overhead. As an example of a UDP-based protocol that has mechanisms integrated into the protocol for reliability and transaction management, we have chosen the Multicast Dissemination Protocol (MDP) [8] for our experiments. MDP was developed to produce reliable communications for multiple users while also promoting efficient utilization. This is done via the use of selective negative acknowledgments, the use of parity for frame repairs, and link management options. This type of protocol design is also expected to be useful in the space channel environment even if a single user, point-to-point link is being used. The MDP protocol does not use a windowing mechanism to control data transmission and avoid congestion. Rather, a *transmit_rate* parameter that can be adjusted in real-time is used to control data flow. Since the typical satellite command or telemetry channel is not congested by other users, the rate control parameter can be set to a value that works best with the transmission hardware. A related UDP-based file transfer protocol is the Consultative Committee for Space Data Systems File Delivery Protocol (CFDP) which is currently under development [7]. At the time of these tests, MDP has the advantage of being more readily available on a larger number of platforms for experimentation than CFDP.

This paper concentrates evaluating the relative performance of MDP over a TCP-based protocol such as *ftp* across a simulated low data rate satellite channel. This satellite channel is configured to emulate a link from a ground station through a relay satellite at geostationary orbit and back to a user satellite at low-earth orbit. The channel error rate will be set to a range of values to be expected with small satellite payloads. The goal of this study is to answer two basic questions about this configuration:

1. is there an overall advantage to MDP over TCP-based file transfer protocols in our simulated satellite channel under normal operating conditions and if there is to quantify the advantage

2. is there an advantage to a MDP in this link environment when link outage problem occur and can the protocol recover.

In this configuration, data transmission rates will be limited to a maximum of 115200 bits per second. The channel bit error rates will range from 10^{-4} to 10^{-6} . We will also look at "error free" channels, that is, channels where the bit error rate is so low that errors typically do not occur during a single file transfer.

In this paper, we will describe the overall simulation environment to model the space channel for the small satellite user. We will also examine the basic TCP/IP baseline performance and the optimal frame size for transmission as a function of channel error rate. Then we will look at the performance of MDP as a function of channel error rate. Finally, we will look at the performance of MDP when the channel is discontinuous such as that found with satellite passes.

SIMULATION ENVIRONMENT

The protocol testing environment consists of a set of computers running software to emulate conditions found in space communications data channels. The channel simulator configuration is illustrated in Figure 1. The user can select the desired data transmission rate, the channel error rates, the channel propagation delay, and channel drop out timing, if required. All of this is done with baseband data sets and not modulated waveforms. The computers are executing a LabVIEW Virtual Instrument (VI) to simulate the channel conditions. Here, T_x is the logical ground station while R_x is the logical satellite. The 1×2 module and the 2×1 module both perform rate changes between the simulator and the source and destination computers. These modules are capable of adding delays and breaks to the forward and return data links, respectively. Here, a forward link is the link from the controlling entity to the subject entity. In a satellite system, this corresponds to the link from the earth station to the satellite and is typically used to send command data. Similarly, a return link is the link from the subject entity to the controlling entity. In a satellite system, this corresponds to the link from the satellite to the earth station and is typically used to send satellite housekeeping data or instrument data.

The protocol testing initiative was developed to provide an independent testbed to evaluate the performance of networking protocols in the space channel environment, especially that found in the small satellite design environment. In particular, we were attempting to address channels with the following characteristics:

1. Forward and return data rates from 2400 bps through 115200 bps, selectable in each direction,
2. Bit Error Rate (BER) selection from 0 through 0.0001, selectable in each direction,
3. Data file sizes from 1 KByte through 1 MByte,
4. Short transfer opportunities, e.g., typical 5-minute ground station passes,
5. User-selectable delay times for the forward and return link up to 5 seconds
6. Channel delays up to 5 seconds in each direction, and
7. Channel outages at user-selectable times and for user-selectable durations.

The details of the simulator are given in [9]. The T_x and R_x computers will be running a version of the Linux operating system with the Point-to-Point Protocol (PPP) as the

channel level frame format. In our investigations, we will examine both the default PPP frame length and a length optimized for the channel BER.

The forward and return links can be configured as either a symmetric or asymmetric dual-simplex links. A symmetric channel is a simulation environment in which the forward link and the return link have the same data rate. An asymmetric channel is a simulation environment in which the forward link and the return link do not have the same data rate. Unless otherwise stated, the forward link will have a smaller data rate than the return link for asymmetric channels.

The data sets used in the protocol testing will be random data files. That is, the file is a set of random data bytes used emulate an actual data source from a sensor. While this is not an exact analog of text file contents, it is believe to be representative of most data transmissions. The testing that is performed on a protocol will typically involve the following measurements:

1. File transmission time: the time from the start of a data transmission until the end of the transmission,
2. Relative channel throughput: this is the file length (bits) divided by the file transmission time and normalized by the channel baseband data rate. The quantity is unitless (bps/bps).

Typical experiments will consist of transmission experiments in a given configuration with a set BER. A total of 16 files of each size will be transmitted in a random order and the time to complete the transmission will be recorded. These experiment runs will be used to determine the significance of differences between pairs of experiment configurations. Using the mean of the first configuration, μ_A , and the mean of the second configuration, μ_B , Tukey's Honestly Significant Difference (HSD) [10] will be used to determine if the null hypothesis, that the means are the same, is to be accepted or rejected. The null hypothesis is rejected if

$$|\mu_A - \mu_B| \geq q_{\alpha, a, a(n-1)} \sqrt{\frac{MS_e}{n}} \quad (1.1)$$

Here, n is the number of measurements made in each experiment, a is the number of measurement sets being compared, $1-\alpha$ is the confidence level (assumed to be 95%), and q is the studentized critical value for the measurement set. For our experimental configuration with 16 runs and comparing two groups, a q of 2.888 is used.

BASELINE PERFORMANCE TESTING

Because of its general familiarity, we will use the TCP/IP file transfer protocol (*ftp*) as the baseline for comparison when looking at the MDP results. We will also use the *ftp* performance to set the optimal frame size for use in the experiments as a function of the channel BER. In Table 1, we list the transfer times and their standard deviation when using the default PPP frame size of 1500 bytes. The optimal frame size in the presence of channel errors is not necessarily the maximum frame size permitted by the protocol. If the frame has a high probability of error, then the retransmissions will override the efficiency of having a greater frame length. A series of *ftp* tests were conducted to determine the optimal frame size, in the sense of minimizing file transmission time, for the channels used in this study. In Table 2, we list the optimal frame size, the average time to complete a file transfer and the standard deviation for the measurements. The

times from Tables 1 and 2 are illustrated in Figures 2 and 3 for the default and optimal frame sizes, respectively. In this baseline experiment, we examine only transfers with 100-kByte and 1-MByte files. The optimal frame size declines with increasing channel error rate. For the 10^{-6} BER case, the difference between the optimal size and the default PPP size of 1500 bytes is not statistically different. Consequently, the 1500-byte size will be used as the optimal in subsequent tests for these low-error channels. For the 10^{-5} BER case, there is a significant reduction in the throughput time when the smaller PPP frame size is used. The results are continued for the 10^{-4} BER case. With this high channel error case and default frame size, the TCP/IP flow control mechanism keeps the file transfers from occurring. By cutting the PPP frame size to 128 bytes, the 100 KB file transfers were able to be completed, albeit at a rate one-fifth those of the zero-error channel case. The 1-MB file transfers did not reliably run to completion so this case would not be expected to be useful in a high-error channel environment. In the MDP testing, we will investigate the file transfer throughput of the MDP protocol and compare that with the *ftp* performance. We will look to see if the PPP frame size affects the results as it does for TCP/IP as well.

MDP FILE TRANSFER TESTING

The MDP testing performed over the simulated space channel is done with the default settings for the MDP protocol just as it was with the default settings for *ftp*. The only MDP environment variable that was set was the *transmit_rate* parameter. Although the computers in the simulator are rated for 115200 bps data transfers, they cannot sustain this rate in continuous transmission. By testing, it was found that this parameter needed to be set to 88 kbps to keep from overflowing the hardware buffers in the interface. This problem was not encountered with the TCP/IP testing where the windowed flow control keeps the sustained rate below the maximum rate on the channel. With UDP-based protocols having no flow control, these buffering considerations need to be taken into account when configuring the link.

The experiment results for MDP using the default PPP frame size and the optimal size obtained with the *ftp* testing are given in Table 3. The throughput times as a function of file size and channel BER are plotted in Figures 4 and 5 for the default and *ftp*-optimal frame sizes, respectively. The effective channel throughput for both cases is plotted in Figures 6 and 7. The first result that was noticed with MDP is that it has much better performance in high BER channels than does a TCP/IP-based protocol.

The first significance test is summarized in Table 4. Here, we compare the two MDP configurations: using the default PPP frame size and the *ftp*-optimized frame size. From Table 3, we see that the default PPP frame size of 1500 bytes gives a shorter file transfer time than does the *ftp*-optimized frame size. For the channels with a BER less than 10^{-6} , the default and optimized frame sizes are the same so we will concentrate on the high-error channels. Based on the HSD threshold computed using Equation (1) for the test results given in Table 4, we see that the advantage to using the default frame size in the high-error channels is statistically significant because the difference in the means exceed the threshold in both cases (the null hypothesis is rejected). This implies that MDP operates better over the PPP link when using the default frame size rather than one tuned for the channel error rate.

The second significance test is summarized in Table 5. Here, we compare the MDP transfer with the default frame size with the *ftp* transfer using the optimized frame size on the PPP link. For the low-error channel cases, *ftp* had superior performance to MDP. While the difference is statistically significant because the difference in the means exceeds the threshold computed using Equation (1), the difference is not large – on the order of 2 to 4 seconds. This is expected, in part, because MDP is running with the *transmit_rate* parameter keeping the data flow from achieving the maximum rate, as explained above. For the high-error channel cases, the trend goes in favor of MDP. For the 10^{-4} channel, the difference between MDP and *ftp* was statistically different because the difference in the means exceeded the threshold and the actual difference was quite large.

Table 6 summarizes the relative performance of *ftp* and MDP in these cases. The *ftp* performance is the worst when the protocol is most apt to confuse channel errors with network congestion and thereby reduce the throughput. The UDP-based MDP protocol is trying a best-effort service in all cases and does not window the data flow, even in high-error channel cases. The persistence appears to be more important than the optimal frame size and flow control in overcoming channel errors.

LINK FAULT TESTING

One of the major problems in satellite communications for low-earth orbit payloads is the short contact time to ground stations for the orbiting satellite. Typical ground station contact times are approximately 5 minutes depending upon the orbital parameters and the ground station location. The terrestrial Internet has nearly 100% connectivity when compared with the ground station connectivity. A second source of link drop outs is from spin modulation in the satellite when a spinning satellite does not point its antenna correctly for the receiving station. The ability of a protocol to recover from transmission gaps of varying durations is valuable to operating the communications link.

The first set of tests run was to cover the case of short gaps that might occur due to spin modulation. In these cases, the gap duration is of the order of magnitude from the expected time for the file transfer to occur based on the *transmit_rate* parameter and the file size. Table 7 shows the results of these tests. The results from the table show that the transmission time is basically extended by the gap duration regardless of where the gap occurs in the transmission. In these cases, MDP is able to recover from the gap and resume the transmission.

When transmissions are interrupted by the loss of signal at a ground station, there may be a gap on the order of the orbital period (or longer) until the transmission can resume. In these cases, the MDP protocol can resume the transmission after a re-synchronization time. Table 8 shows the necessary resynchronization time in the MDP default configuration. From this, we can see that there is some delay in resuming the transmission but the transmission does resume and run to completion.

CONCLUSIONS

In this paper we have examined the suitability of using the Multicast Dissemination Protocol in an environment that would be expected to be found in a small satellite communications environment. These environments have limited data rates, high channel error rates, and gaps in the transmission. We have used the TCP/IP protocol

ftp as the basis for comparison of the file transfer times. In this series of experiments, we have seen that

1. The ftp protocol is sensitive to the link layer framing protocol values. In particular, link frame size should be reduced for high error rate channels.
2. The MDP protocol works better with the default PPP frame size of 1500 bytes than by using smaller frame sizes in high-error channels as should be used with ftp.
3. The ftp and MDP protocols work similarly for low-error-rate channels. The differences are small but statistically significant.
4. MDP works better than ftp for high-error-rate channels. The MDP transfer times can be considerably smaller for MDP than ftp in these cases.
5. MDP can survive transmission gaps and re-start the data transfer even after a long gap.

From this performance, MDP is a viable candidate for use in the small satellite environment. The data throughput performance is nearly the same as ftp for high quality channels. For low quality channels, MDP is better than ftp. From an operational consideration, the value of being able to switch from one protocol to another based on momentary channel changes is questionable so MDP would seem to be a reasonable choice over a wide range of channel conditions. The ability to resume transmissions after gaps is also important for satellite communications with limited ground station contact time. In this case, MDP shows advantages over ftp.

REFERENCES

- [1] Mark Allman, Chris Heyes, Hans Kruse, Shawn Osterman, "TCP Performance over Satellite Links," Proc. 5th International Conf. on Telecom. Systems, 1997.
- [2] Yongguang Zhang, Elliot Yan, Son K. Dao, "A Measurement of TCP over Long-Delay Network," Proc. 6th International Conf. on Telecom. Systems, March 1998.
- [3] Ian F. Akyildiz, G. Morabito and S. Palazzo Member, "TCP Peach: A New Congestion Control Scheme for Satellite IP Networks," *IEEE Transactions on Networking*, Vol 9, No. 3, June 2001, p. 307.
- [4] K. Bhasin, and J. L. Hayden, "Space Internet Architectures and Technologies for NASA Enterprises," *Proceedings 2001 IEEE Aerospace Conference*, paper 4.0201, March 2001.
- [5] E. Criscuolo, K. Hogue, and R. Parise, "Transport Protocols and Applications for Internet Use in Space," *Proceedings 2001 IEEE Aerospace Conference*, paper 4.0203, March 2001.
- [6] R. Durst, G. Miller, and E. Travis, "TCP Extensions for Space Communications," *2nd ACM International Conference on Mobile Computing and Networking*, November 1996, p. 15.
- [7] Burleigh, S., et al., "Delay-Tolerant Networking: An Approach to Interplanetary Internet," *IEEE Communications Magazine*, Vol. 41, No. 6, June 2003, p. 128.
- [8] Macker, J.P. and Adamson, R.B., "The Multicast Dissemination Protocol (MDP) Toolkit," MILCOM 2000, Los Angeles, October 2000.
- [9] Horan, S. and Wang, R., "Design of a Space Channel Simulator Using Virtual Instrumentation Software", *IEEE Transactions Instrument and Measurements*, Vol. 51, No. 5, October 2002, p. 912-916.

[10] Dowdy, S. and Wearden, S., *Statistics for Research*, 2nd ed., Wiley: New York, 1991, pp. 302 – 312.

Table 1 –ftp Transmission Using Default PPP Frame Size				
	Channel BER			
File Size	0	0.000001	0.00001	0.0001
1 MByte				
Mean Transfer Time (seconds)	91.53	96.90	217	Did not reliably complete transfer
Standard Deviation (seconds)	0.29	1.60	21.4	
100 kByte				
Mean Transfer Time (seconds)	7.40	8.01	22.0	Did not reliably complete transfer
Standard Deviation (seconds)	0.13	0.47	9.11	

Table 2 – Optimal PPP Frame Size from ftp Transmission				
	Channel BER			
File Size	0	0.000001	0.00001	0.0001
1 MByte				
Optimal Frame Size	1500 B	1450 B	650 B	128 B
Mean Transfer Time (seconds)	91.5	96.0	139	Did not reliably complete transfer
Standard Deviation (seconds)	0.29	0.98	5.4	
100 kByte				
Optimal Frame Size	1500 B	1400 B	650 B	128 B
Mean Transfer Time (seconds)	7.40	7.99	11.8	339
Standard Deviation (seconds)	0.13	0.56	1.02	169

Table 3 – Average Times in Seconds for File Transmission Using MDP				
	Channel BER			
File Size	0	0.000001	0.00001	0.0001
10 MByte				
Default Frame Size	953.38	1025.69	1118.81	3220.50
Optimal Frame Size	953.38	1025.69	1206.94	Excessive duration
1 MByte				
Default Frame Size	95.31	99.00	107.44	321.56
Optimal Frame Size	95.31	99.00	117.81	2338.31
100 kByte				
Default Frame Size	9.63	10.88	11.56	38.00
Optimal Frame Size	9.63	10.88	13.63	145.00
10 kByte				
Default Frame Size	1.00	1.25	1.88	7.31
Optimal Frame Size	1.00	1.25	3.00	15.50

Table 4 – Significance Test for Differences Between MDP with Default Frame Size and t_{tp} -optimal Frame Size			
File Size			
1 MByte	BER	Threshold	$\mu_A - \mu_B$
	0	0.35	0.00
	0.000001	0.65	0.00
	0.00001	1.11	10.38
	0.0001	52.1	2017
100 kByte	BER	Threshold	$\mu_A - \mu_B$
	0	0.36	0.00
	0.000001	0.45	0.00
	0.00001	1.07	2.06
	0.0001	9.87	107.00
10 kByte	BER	Threshold	$\mu_A - \mu_B$
	0	0.00	0.00
	0.000001	0.62	0.00
	0.00001	0.85	1.13
	0.0001	2.38	8.19

Table 5 – Significance Test for Differences Between MDP with Default Frame Size and t_p Using the Optimal Frame Size			
File Size			
1 MByte	BER	Threshold	$ \mu_A - \mu_B $
	0	0.29	3.79
	0.000001	0.68	2.96
	0.00001	2.78	31.88
100 kByte	BER	Threshold	$ \mu_A - \mu_B $
	0	0.26	2.22
	0.000001	0.43	2.88
	0.00001	0.64	0.26
	0.0001	86.29	301.06

Table 6 – Best Protocol Choice for Channel BER and File Size		
BER	1 MByte File	100 kByte File
0	ftp	ftp
0.000001	ftp	ftp
0.00001	MDP	No difference
0.0001	MDP	MDP

Table 7 – Effect of Link Cuts on Transfer Duration for Short Gaps			
File Size	Cut at % of Transmission Time	Gap Duration (seconds)	Total Transmission Time (seconds)
1 MByte	25	15	110
1 MByte	50	15	110
1 MByte	75	15	110
1 MByte	25	30	126
1 MByte	50	25	120
10 MByte	50	120	1076

Table 8 – Resynchronization times for Scheduled Link Cuts		
Cut at % of Expected Transmission Time	Gap Duration (seconds)	Resynchronization Time (seconds)
25	900	120-159
25	1800	105 – 158
25	5400	150
50	5400	150 – 165

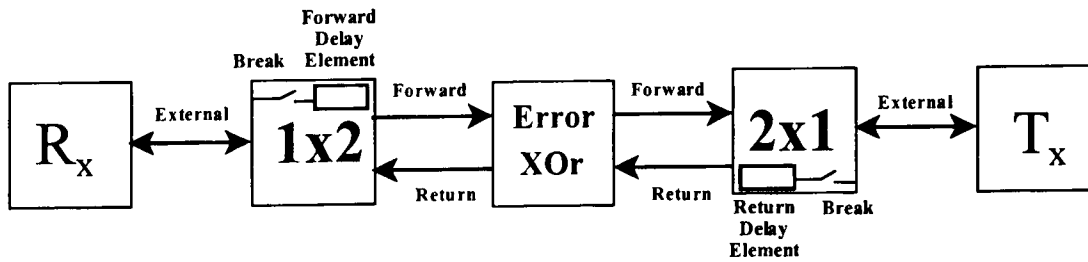


Figure 1 - Space channel simulator for modeling errors and delays.

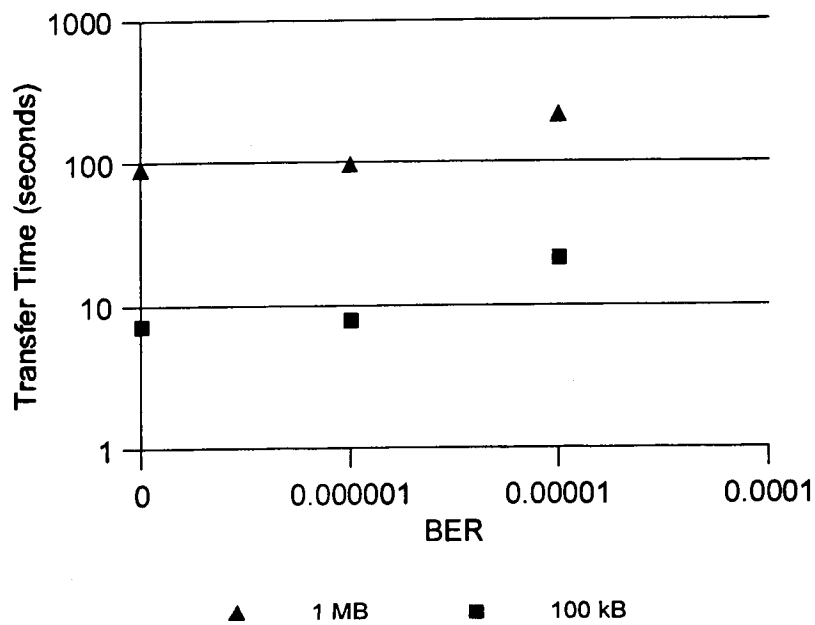


Figure 2 -- File transfer times for the ftp protocol using the default PPP frame size as a function of file size and channel error rate.

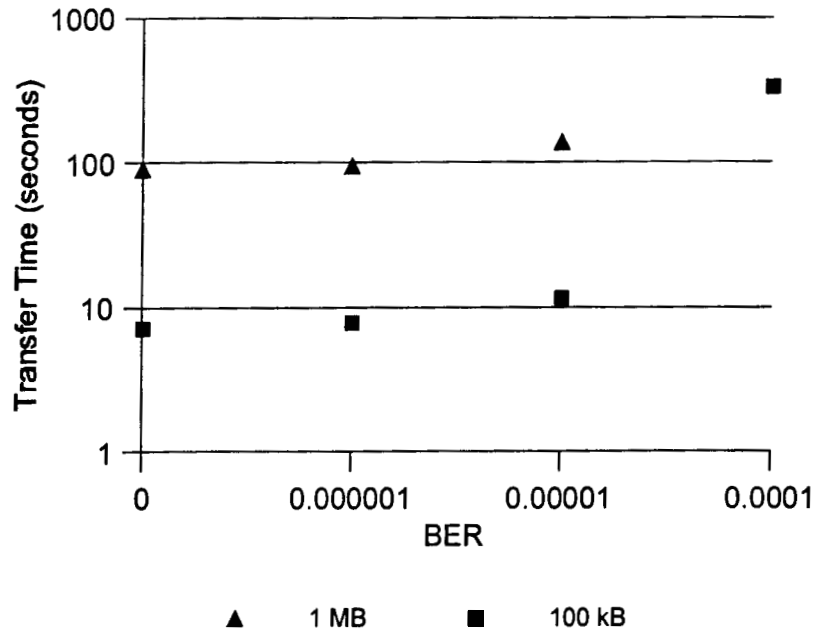


Figure 3 -- File transfer times for the ftp protocol using the optimal PPP frame size as a function of file size and channel error rate.

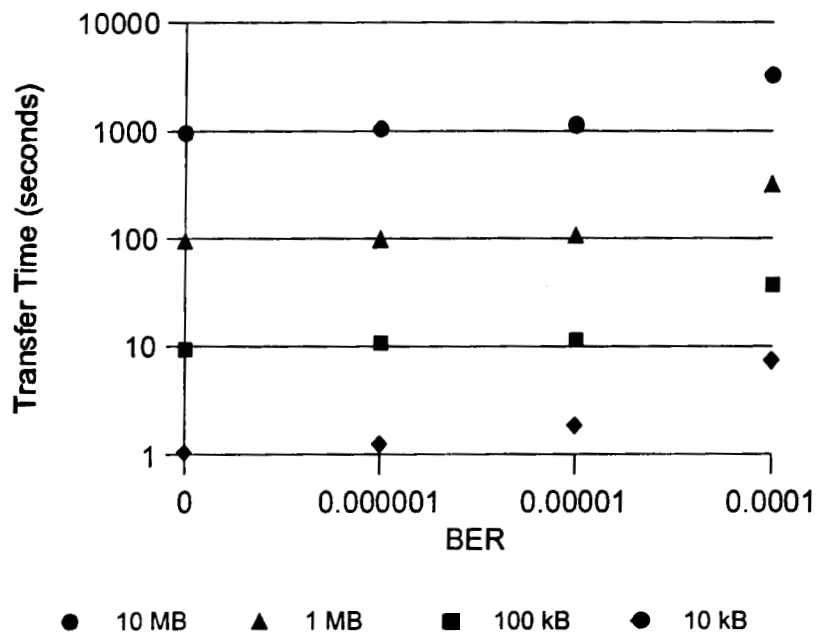


Figure 4 -- File transfer times for the MDP protocol as a function of channel BER and file size using default PPP frame size.

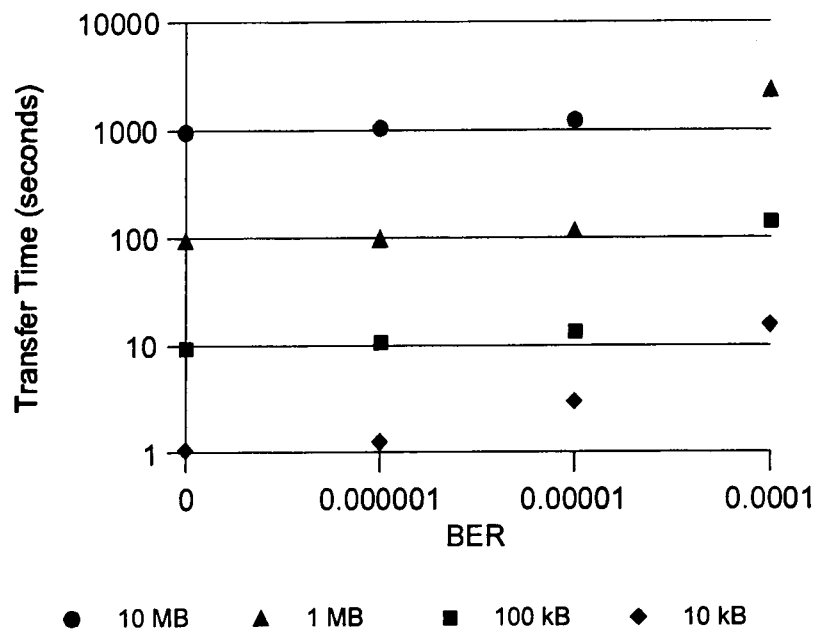


Figure 5 – MDP Transfer time as a function of channel BER and file size using ftp-optimized PPP frame size.

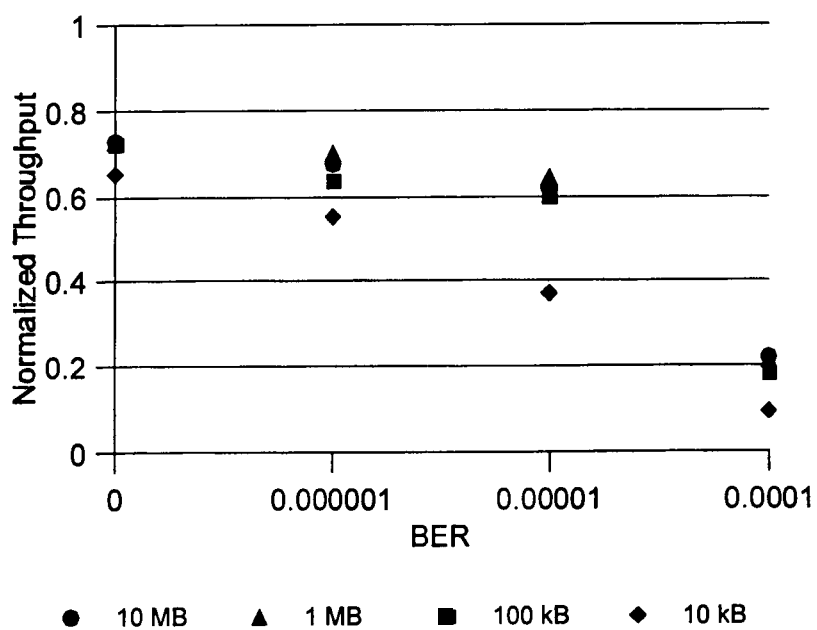


Figure 6 -- MDP throughput as a function of channel BER and file size for the default PPP frame size.

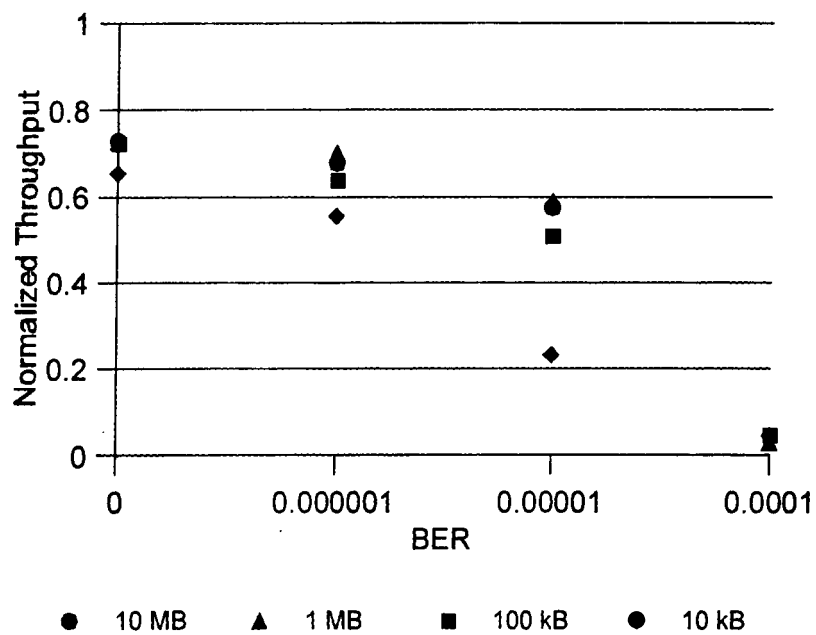


Figure 7 -- MDP throughput as a function of channel BER and file size for the ftp-optimal PPP frame size.

5 Autonomous Reconfiguration of Ground station Receivers

A new effort was started this year to consider how signal processing techniques could be applied to the problem of determining signal data rate, bit encoding, and convolutional encoding method from just the baseband signal alone. The following paper was submitted to the 2003 International Telemetry Conference and it summarizes the results to date in attacking this problem.

A DESIGN FOR SATELLITE GROUND STATION RECEIVER AUTOCONFIGURATION

Phillip De Leon, Qingsong Wang, Steve Horan, Ray Lyman
New Mexico State University
Klipsch School of Electrical and Computer Engineering
Las Cruces, NM 88003
{pdeleon, qwang, shoran, rlyman}@nmsu.edu

ABSTRACT

In this paper, we propose a receiver design for satellite ground station use which can demodulate a waveform without specific knowledge of the data rate, convolutional code rate, or line code used. Several assumptions, consistent with the Space Network operating environment, are made including only certain data rates, convolutional code rates and generator polynomials, and types of line encoders. Despite the assumptions, a wide class of digital signaling (covering most of what might be seen at a ground station receiver) is captured. The approach uses standard signal processing techniques to identify data rate and line encoder class and a look up table with coded sync words (a standard feature of telemetry data frame header) in order to identify the key parameters. As our research has shown, the leading bits of the received coded frame can be used to uniquely identify the parameters. With proper identification, a basic receiver autoconfiguration sequence (data rate, line decoder, convolutional decoder) may be constructed.

KEY WORDS

Digital receiver, "smart" receiver, ground station receiver, autoconfiguration, self-configuration

Introduction

Users of NASA's Space Network (SN) Multiple Access service may transmit their data at different data rates, i.e. 9600bps, 1Mbps, etc.; may convolutionally encode their data at different rates, i.e. rate 1/2, rate 1/3, etc.; and may line encode their data with one of several standards, i.e. Not Return to Zero (NRZ) or Biphasic (BiΦ) also known as Manchester encoding [1], [2]. Normally, these communications parameters are known in advance so that the Ground Station Receiver (GSR) can be configured. The question we address in this paper is whether from the data signal itself, these parameters (data

rate, convolutional encoder rate, line encoder) can be identified. Our motivation is to design a receiver which can configure itself automatically (autoconfigure).

In addition, a GSR autoconfiguration would be useful in the event of an emergency onboard a spacecraft. In such an event, the spacecraft might enter into alternate operating modes usually varying data communications parameters in an attempt to reestablish communications. With such an autoconfiguration receiver, ground station controllers could be aided in fault recovery. As an example, with the Hubble Space Telescope (HST), both the data rate and data format change when it goes into a "safe-hold." This event leads to a loss of data until controllers have realized HST is in the safe-hold condition [3]. Finally, such an autoconfiguration feature might be useful in forming clusters of autonomous, heterogeneous satellites where communications systems will need to adapt to different techniques in real-time.

As a first approach toward estimating the parameters, one might consider discriminating temporal, spectral, or statistical characteristics of the various codes (convolution and line) and make use of these characteristics to estimate the parameters. However, other than using the DC response to estimate the line code class (NRZ or BiΦ) but not the particular format (Line, Mark, or Space) or convolutional code rate, it is not clear this approach can yield information useful in parameter estimation. As a second approach, we consider the use of standard telemetry frame header bits used in synchronization tasks. We will generically call these header bits, sync words. The idea is that these sync words when coded, yield unique information sufficient to estimate the parameters. Of course this uniqueness depends on significant reduction of the number of sync words, code rates, and data formats under consideration from the state space of all possible combinations that can be envisioned.

In the next subsections, we give a brief review of the encoder structures found in the Space Network User's Guide (SNUG). We consider only rate 1/2 and 1/3 convolutional coding as well as uncoded data and six standard line codes (NRZ-L, -M, -S; BiΦ-L, -M, -S) used in most telemetry applications. In the remainder of the paper, we describe the use of the sync words in estimating the key parameters. We also present results concerning the accuracy of these estimates.

Convolutional Coder

Fig. 1 shows a constraint length $M+1$, rate 1/2 convolutional coder. Here the generator polynomial for the i^{th} path from input to output is

$$g^{(i)}(D) = g_0^{(i)} + g_1^{(i)}D + g_2^{(i)}D^2 + \dots + g_M^{(i)}D^M \quad (1)$$

where D represents a unit delay [4]. A rate 1/3 coder would have an additional path from input to output. As described in the SNUG, only convolutional code rates 1/2 and 1/3 are used, each with constraint length 7. In addition, the generator polynomials under consideration as given in the SNUG are [1]:

$$\begin{aligned}
g^{(1)}(D) &= g_0^{(1)} + g_1^{(1)}D + g_2^{(1)}D^2 + g_3^{(1)}D^3 + g_6^{(1)}D^6 \\
g^{(2)}(D) &= g_0^{(2)} + g_2^{(2)}D^2 + g_3^{(2)}D^3 + g_5^{(2)}D^5 + g_6^{(2)}D^6 \\
g^{(3)}(D) &= g_0^{(3)} + g_1^{(3)}D + g_2^{(3)}D^2 + g_4^{(3)}D^4 + g_5^{(3)}D^5 + g_6^{(3)}D^6
\end{aligned} \tag{2}$$

[The third polynomial, $g^{(3)}(D)$, is obviously not used in the rate 1/2 encoder]. Finally, uncoded (straight-through path from input to output) data may also be used [1].

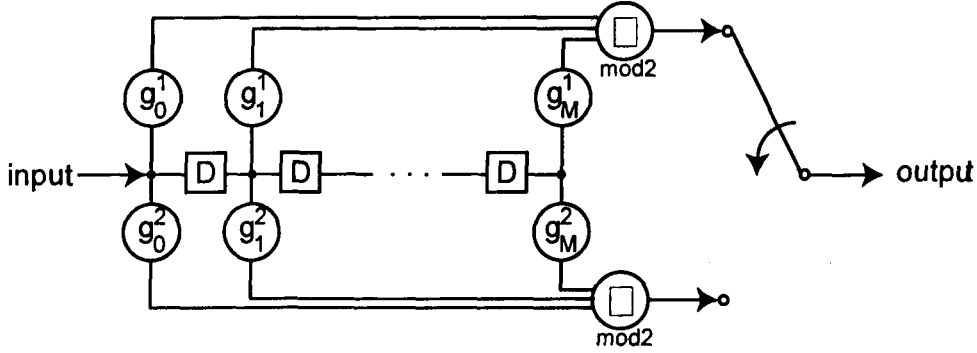


Figure 1: Constraint length M , rate 1/2 convolution coder

Line Coder

Prior to transmission, the data stream is converted to a waveform and the logical zeros and ones are encoded within this waveform as high (H) or low (L) voltage levels. The waveform or line encoding is used to assist in clock extraction at the receiver as well as to shape the spectrum of the transmitted signal [5], [6]. The usual desired spectral properties include a zero DC response and fast rolloff of the high-frequency spectrum. In this work, we consider the two line code classes which are standard for space telemetry: NRZ or BiΦ. For BiΦ, we consider the IRIG-106 definitions [4], [5]. Within each line code class are various formats: Line (L), Mark (M), and Space (S). The basic definitions are summarized in Table 1 below [2].

Table 1: Line Encoder Definitions

NRZ-L	"1", "0" is represented by a high (H), low (L) level during the symbol period, respectively.
NRZ-M	"1" is represented by a change in level. "0" is represented by no change in level.
NRZ-S	"0" is represented by a change in level. "1" is represented by no change in level.
BiΦ-L	"1" is represented by a midperiod transition from H to L. "0" is represented by a midperiod transition from L to H.
BiΦ-M <i>IRIG-106 Definition</i>	"1" is represented by a midperiod transition. "0" is represented by no midperiod transition.
BiΦ-S <i>IRIG-106 Definition</i>	"0" is represented by a midperiod transition. "1" is represented by no midperiod transition.

Overall Encoding System

Fig. 2 illustrates the basic operation of the telemetry frame encoding process. A frame is assembled with a sync word in the header followed by data. This frame is then passed through the convolutional encoder and line coder resulting in a coded frame. The goal of the autoconfiguration receiver is to estimate the convolutional code rate and line code class parameters given only the coded frame. We will not assume any knowledge of the length of the sync word, thus we are not able to establish boundaries within the coded frame for where the coded sync word ends and where the coded data begins.

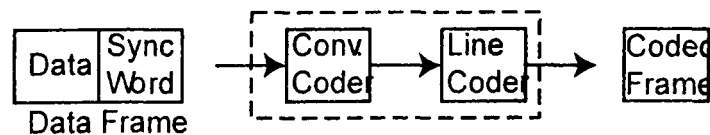


Figure 2: Overall coding procedure. Telemetry frame composed of a sync word (header) with a data payload. After convolutional and line coding, the result is a coded frame.

Receiver Design Approach

The proposed approach toward a design for receiver autoconfiguration involves two stages as illustrated in Fig. 3. The first stage estimates data rate and the line code class (either NRZ or BiΦ). The second stage involves a look up table (LUT) whose entries or keys contain the coded sync words. These keys can be generated by passing all sync words under consideration through the structure in Fig. 2 for all combinations of convolutional encoder and line encoder that are to be considered. The LUT is constructed prior to the autoconfiguration and may be stored in a ROM.

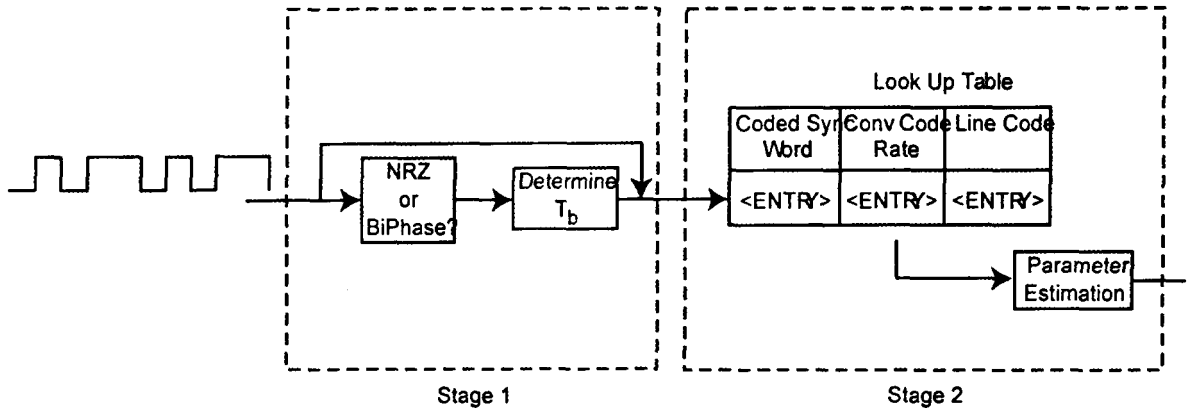


Figure 3: Proposed design for receiver autoconfiguration

In order to establish the data rate, we must know whether the line code is NRZ or BiΦ since the latter involves a voltage transition mid-period which would otherwise be interpreted as a doubling of the bit rate. In addition, once the line code class is established, the search space within the LUT can be reduced by half since only those keys with the line code need be searched.

Stage 1: Data Rate, Line Code Class Estimation

We assume a spread spectrum system, knowledge of a user's Pseudo Noise (PN) code, and PN lock up to the user have already been accomplished by ground station operations. We assume at the input a despread, PCM signal (waveform) resembling Fig. 4 is available for analysis. For now, we do not assume any pulse shaping, e.g., raised cosine filtering, has been applied to the bits. If this has been applied, we assume that the bits have been processed to yield a waveform with rectangular pulses. In order to estimate the data rate, we propose to oversample the waveform for a short period of time and measure the smallest width pulse. Assuming a sample rate of f_s , and a shortest pulse width of L samples, then an estimate for the bit period is given by

$$\hat{T}_b = cLf_s \quad (1)$$

where c is 1 if NRZ line encoding is used and 2 if Bi Φ is used [actual identification of the line code format (L, M, or S) is handled by Stage 2]. The estimate for the data rate is then

$$\hat{R}_b = \frac{1}{T_b}. \quad (2)$$

Once c is known, this estimate can be compared to a list of “most likely” data rates with the closest match selected. This provides the data rate, R_b . It should be noted that in an effort to reduce cost, more and more space communication system designs use commercial off-the-shelf chip sets. These chips normally implement CCITT standards for which only certain data rates are allowed. As a start, the allowable data rates would form the list of “most likely” data rates which the proposed receiver design would compare against.



Figure 4: Example waveform.

From the waveform, we must now determine what line code format is used assuming either NRZ or Bi Φ . There are two approaches to identifying the line code class: frequency-domain and time-domain based. The former is, in theory, more accurate but at a relatively high implementation cost while the latter's implementation is simple but the accuracy is a function of the length of the observed waveform.

Frequency-Domain Approach

By computing the power spectrum, $S(\omega)$, of the waveform one can differentiate between NRZ and Bi Φ since the DC response of Bi Φ is zero while that of NRZ is a peak. Therefore a simple procedure for identifying the line code (either NRZ or Bi Φ) is to compute the power spectrum for the waveform and evaluate the derivative of the PSD at zero. If the derivative is negative (spectrum slopes down from 0Hz) the line code class is NRZ, otherwise the derivative is positive (spectrum slopes up from 0Hz) and the line code class is Bi Φ . The derivative can be approximated by a simple backward difference equation

$$m = S(1) - S(0) \quad (3)$$

where $S(k)$ is the k th point in the spectrum. This procedure requires an FFT calculation and evaluation of (3).

This approach has the disadvantage of taking a relatively long period to compute not to mention the computational requirements. In the next subsection, we investigate a simpler approach that yields equivalent results.

Time-Domain Approach

The second approach to identify the line code class is strictly time-domain based. With this method we exploit the fact that with BiΦ, we cannot have a pulse width more than $1.5\hat{T}_b$ since each bit must have a transition. Therefore, we need only examine the waveform and check to see if there exists a pulse width with duration exceeding $1.5\hat{T}_b$. If such a pulse exists, the line code is NRZ otherwise it could be either NRZ or BiΦ. Since it is unreasonable to examine the entire waveform, we must determine how much of the waveform one would need to examine to be within some desired probability of correctly identifying the line code. Assuming +V and -V are equally likely, Fig. 5 illustrates how many bit periods one needs to examine to have a probability, P that no pulse of duration $1.5\hat{T}_b$ exists. We can show that to be 99.99% certain there is no pulse of width $1.5\hat{T}_b$, we must examine 48 bits. If during the examination, we encounter a pulse of width more than $1.5\hat{T}_b$ we are certain that NRZ is the line code and the procedure can stop. Otherwise we would default to BiΦ. The implementation for such an approach involves a simple digital filter (assuming ± 1 levels) and is shown in Fig. 6.

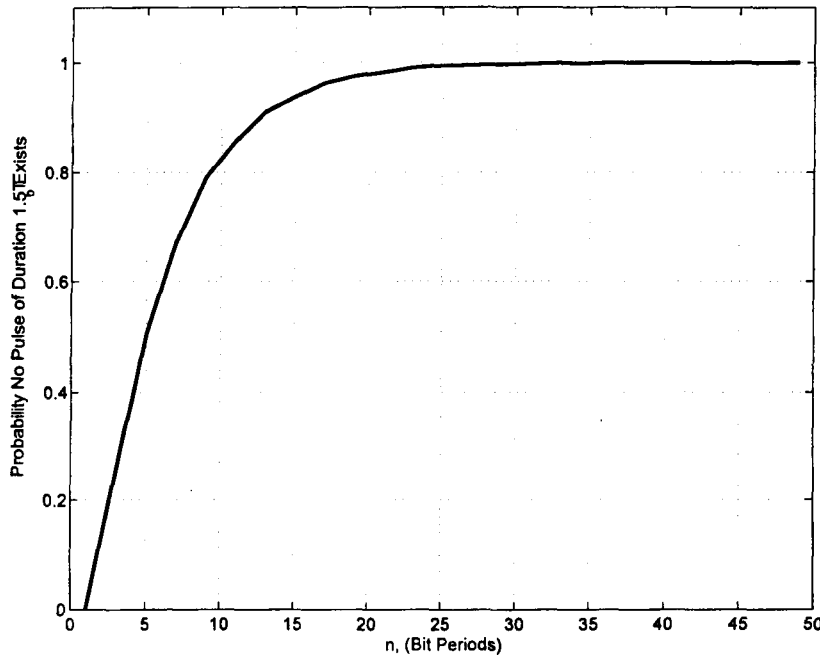


Figure 5: Probability of no pulse width more than $1.5\hat{T}_b$ as a function of the number of bits examined.

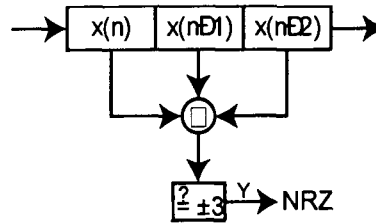


Figure 6: Time-domain approach to line code class identification.

Stage 2: Look Up Table

In order to determine the particular line code (-L, -M, -S) and coding rate, we make use of what is known about the structure of the transmission frames. These are always preceded by a sequence of known symbols to aid the receiver in synchronization tasks. We use the generic term *sync word* for these known sequences.

The sync word used is specified in the data-link protocol. Users of the SN most often employ HDLC or one of the CCSDS recommendations. The most commonly used sync words from these are listed in Appendix A. Even if the protocol in use is not known in advance, if the set of possible sync words is small enough, then we may use a LUT approach to determine the code rate and line code.

The LUT is constructed by convolutionally encoding each of the sync words at each coding rate of interest. Then, each resulting sequence is line coded using each line code of interest. The resulting output sequences become keys in the LUT. When a frame is received, we check the first N binary symbols of the frame against the keys in the LUT, and when a match is found, we choose the line code and coding rate corresponding to that key. As in Fig. 7, we find that as long as we choose N greater than or equal to 13, there is no chance that identical keys will yield different values for the coding rate or line code. It is likely that when a frame is first received, we will not be examining the header of the frame. Therefore it is proposed that the above procedure is repeated, shifting in one new symbol until the match is detected thereby making a running correlator as typically found in frame synchronization. Work is in progress in quantifying the impact of this initialization.

Let us consider a couple of examples. Since the sixth and eighth sync words from Appendix A are identical in the first 96 bits, it is clear that, after convolutional encoding and line coding, the resulting output sequences will be identical in the first 13 binary symbols. But this causes no ambiguity because we find that any pair of identical output sequences always corresponds to the same transmission parameters; i.e., the same code rate and the same line code.

On the other hand, suppose we consider only the first $N = 12$ received binary symbols. We find that the fifth sync word from Appendix A produces output sequences which are identical in the first 12 binary symbols when it is convolutionally encoded at rates 1/2

and 1/3. Thus, if these 12 symbols are used as a key in the LUT, it will be impossible to determine which coding rate was applied.

Even with N sufficiently large, one possible difficulty with this approach is in the case of a key with length $K < N$. When comparing a received sequence to such a key, we use only the first K symbols. A problem occurs, though, when the K symbols of this sequence are identical to the first K symbols of another longer key somewhere else on the LUT. To see this, consider the sync word which results in the K -length output sequence. The user data that follows this sync word may cause the first N received symbols to be identical to the longer key. Fortunately, all the sync words of interest, except the HDLC "flag byte" 0 x 7E are longer than 13 bits, and none of the output sequences resulting from the HDLC flag byte exhibit the problem just described.

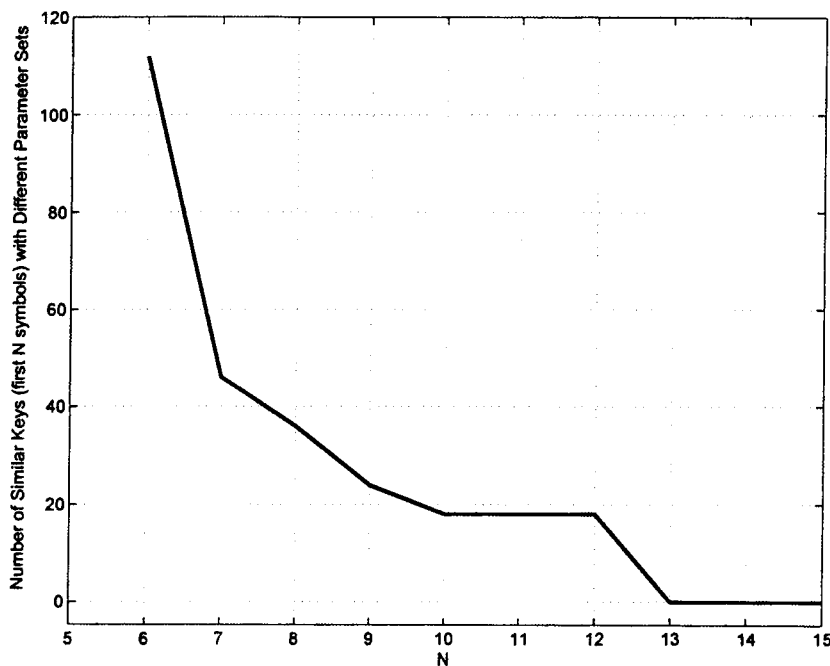


Figure 7: Number of key entries with the same N leading symbols but different parameters

If we consider the 8 sync words in Appendix A, two convolutional code rates (1/2 and 1/3) as well as uncoded data, and six line encoders, an 8K ROM will be sufficient to store the table. We note that both the line decoders and convolutional decoder are implemented with simple digital circuits. Together with the ROM, the system could easily be implemented in an FPGA and interfaced with a GSR.

Future Work

Regarding future work, our immediate task is relaxing the assumption that we know beforehand where the frame begins. Problems to be dealt with include the possibility that a sequence of symbols from mid frame falsely match one of the output sequences in the LUT. We also wish to investigate the impact of physical-layer effects, such as noise, pulse shaping and timing jitter, on our ability to identify the transmission parameters. More long term, we wish to find a more general characterization of the conditions under which this identification can be made. For example, what must be true about the set of sync words, or the parameters to be identified? Such knowledge would allow the techniques developed here to be applied more generally.

Conclusions

In this paper, we have proposed a ground station receiver which autoconfigures itself in the sense that the data rate, convolutional code rate, and line encoder type parameters are all estimated from the received waveform itself. The approach assumes the presence of a sync word in the frame header. After passing the sync word through the convolutional and line encoders, a key is produced and stored in a lookup table. When a frame is received at the receiver, we check the first N binary symbols of the frame against the keys in the table, and when a match is found, choose the coding rate and line code corresponding to that key. Our research has demonstrated that with $N = 13$, there is no chance that identical keys will yield different values for the coding rate or line code.

References

- [1] "Data Encoders," *Space Network User's Guide, Revision 8*, June 2002, Appendix B, p. B-15.
- [2] "Digital Signal Formats," *Space Network User's Guide, Revision 8*, June 2002, Appendix B, p. B-2.
- [3] Dave Israel, NASA Goddard Space Flight Center, private communication, Feb. 2003.
- [4] S. Haykin, *Communication Systems 4th Ed.*, John Wiley & Sons, Inc., New York, NY, 2001.
- [5] S. Horan, *Introduction to PCM Telemetry Systems 2nd Ed.*, CRC Press, 2002.
- [6] Telemetry Group, Range Commander's Council, Telemetry Standards, IRIG Standard 106-01, Part 1, WSMRi, NM, Feb. 2001, p. C-5.

Appendix A: Sync Words Under Consideration

The following table lists the sync words most commonly used on the Space Network.

Table A.1: Commonly Used Sync words, in Hex Notation.

<i>Sync word</i>	<i>Comments</i>
1: 7E	HDLC flag byte
2: EB90	CCSDS: TC
3: FAF320	CCSDS: Proximity-1
4: 1ACFFC1D	CCSDS: TM
5: 034776C7272895B0	CCSDS: TM rate-1/2 turbo
6: 25D5C0CE8990F6C9461BF79C	CCSDS: TM rate-1/3 turbo
7: 034776C7272895B0FCB88938D8D76A4F	CCSDS: TM rate-1/4 turbo
8: 25D5C0CE8990F6C9461BF79CDA2A3F31 766F0936B9E40863	CCSDS: TM rate-1/6 turbo

Appendix B: Sample Lookup Table Entry

The following is a sample entry for IRIG Code 0xFAF320 convolutionally encoded at rate 1/2 and line encoded with NRZ-L.

Coded Sync word	Code Rate	Line Code
...		
...		
1100 0000 1111 0000 0000 1100 0000 1100 1111 0011 0000 0011 0000 0000 1111 1100 1100 1100 0011 1100 0000 0011 0011 0000	1 / 2	NRZ - L
...		
...		

6 Satellite Cluster Communications

A special study was conducted to investigate the potential for commercially-available wireless communications technology to become the basis for the communications systems in clusters of satellites. This study is primarily concerned with communications link analysis and not related topics such as satellite structure or the specific mission needs. There are a large number of potential missions for using these architectural concepts. However, the analysis is not intended to service any particular mission but Low-Earth Orbit missions in general. The full details of the analysis and the spreadsheets used to perform the analysis are given in the technical report "Satellite Cluster Communications" that was submitted earlier.

The study concluded that radio design built around the use of existing commercial wireless transmission techniques has the potential to service as the core technology for a variety of payload classes. Certainly, these can be used in picosatellite and nanosatellites. They become more problematic when moving to systems requiring long ranges and high data rates. In this study, we have sized the antenna and transmission techniques for several mission classes. This analysis can be extended to other mission classes with the application of appropriate parameters.

Recently, this technical report was shared with Space Development Corporation to assist them in concept development for their projects.

7 Bandwidth-Efficient Modulation

The bandwidth-efficient modulation task has been investigating shaped QPSK modulation techniques to assist in reducing the occupied bandwidth of transmissions and still maintain acceptable data quality of service. The following paper was presented at the 2002 International Telemetry Conference and it summarizes the results to date in this study.

PERFORMANCE STUDY OF ENHANCED FQPSK AND CONSTRAINED ENVELOPE MODULATION TECHNIQUES

Deva K. Borah and Stephen Horan
Klipsch School of Electrical & Computer Engineering
New Mexico State University
Las Cruces, NM 88003, USA

ABSTRACT

This paper investigates the spectral properties and the bit error rate (BER) performance of enhanced FQPSK (EFQPSK) and constrained envelope modulation (CEM) techniques. Both the techniques are found to provide good spectral efficiencies. The EFQPSK signals are found to generate spectral lines for unbalanced data. An analytical spectral study for the spectral lines is presented. While the performance of CEM techniques has been presented in [6] for an ideal nonlinear amplifier, we present results for more realistic amplifiers with AM/AM and AM/PM effects. It is shown that such an amplifier generates spectral regrowth and a predistorter is required to reduce the adverse effects. A BER performance study with/without channel coding is also presented for the two techniques.

KEY WORDS

EFQPSK, Bandwidth Efficient Modulation, Nonlinear Amplifier, and Constant Envelope Modulation.

INTRODUCTION

Due to the growing demand on high data rate transmissions in the face of decreasing frequency allocations, bandwidth efficient modulation techniques have received considerable attention in recent times. Both linear and nonlinear modulation techniques have been extensively investigated in the literature. While linearly modulated signals are highly bandwidth efficient in linear channels, their spectral efficiency degrades significantly when power efficient nonlinear amplifiers are used. A nonlinearly modulated signal, such as the Gaussian Minimum Shift Keying (GMSK), can pass through nonlinear amplifiers without spectrum regeneration and performance degradation. However, its spectrum efficiency is low. Therefore, a large number of techniques have been developed in recent times to achieve better spectrum utilization, yet with little performance degradation under nonlinear amplification. The Enhanced Feher-Patented Quadrature Phase-Shift-Keying (EFQPSK) and constrained envelope modulation for linearly modulated signals are two such techniques.

EFQPSK [1] [2] is an enhanced version of Feher's QPSK [3]. This technique emphasizes on symbol by symbol representation of the cross correlation operation. As a result, instead of the crosscorrelator of the conventional FQPSK, the EFQPSK can be described directly in terms of data transitions on the in-phase (I) and quadrature (Q) channels. The EFQPSK also improves upon the smoothness of the modulating waveforms, thus improving the power spectral density roll off. Further, the BER performance has also been significantly improved by exploiting the correlation inherent in the modulation.

The CEM technique has been investigated by several authors [4], [5], [6], [7]. In this technique, a band-limited correcting function is added to a linearly modulated signal in order to constrain its envelope fluctuations. This correction signal limits the signal values at the periodically selected positions. However, it causes the signal to exceed the amplitude threshold at other positions. The CEM technique has been applied to orthogonal frequency division multiplexing (OFDM) in [4], [5]. In [6] and [7], it is used for conventional PSK and APSK signals, and it is shown that the back-off level of the nonlinear amplifier can be reduced, thus obtaining efficient amplification without sacrificing the spectral quality.

In this paper, we investigate the spectral properties and the BER performance of EFQPSK and CEM techniques. Both balanced and unbalanced data are considered. It is observed that EFQPSK signals generate spectral lines for unbalanced data. An analytical spectral study for the spectral lines is presented. While the performance of CEM techniques has been presented in [6] for an ideal nonlinear amplifier, we present results for more realistic amplifiers with AM/AM and AM/PM effects [8]. It is shown that such an amplifier generates spectral regrowth for CEM techniques and a predistorter is required. A BER performance study with/without channel coding is presented for the two techniques.

SYSTEM MODEL

Enhanced FQPSK Modulation: The baseband signal is constructed using eight waveforms $s_i(t)$, $0 \leq i \leq 7$, described in [1]. Each waveform $s_i(t)$ occupies only one symbol interval. So, in every symbol interval, one waveform is chosen for the I channel and another waveform is chosen for the Q channel. The selection of an I/Q waveform depends on the most recent data transition on that channel as well as two most recent successive transitions on the other channel. The I channel baseband waveform $x_I(t)$ during the k -th signaling interval $(k - 1/2)T \leq t \leq (k + 1/2)T$ is given by $x_I(t) = a_{I,k}s_m(t - kT)$, where $a_{I,k}$ is the k -th I channel bit (± 1), and the waveform identification number m is obtained as

$$m = 4 \times |(a_{I,k} - a_{I,k-1})/2| + 2 \times |(a_{Q,k-1} - a_{Q,k-2})/2| + |(a_{Q,k} - a_{Q,k-1})/2| \quad (1)$$

Similarly, the Q channel baseband waveform $x_Q(t)$ during the k -th signaling interval $kT \leq t \leq (k + 1)T$ is given by $x_Q(t) = a_{Q,k}s_n(t - kT)$, where n is obtained as

$$n = 4 \times |(a_{Q,k} - a_{Q,k-1})/2| + 2 \times |(a_{I,k} - a_{I,k-1})/2| + |(a_{I,k+1} - a_{I,k})/2| \quad (2)$$

The overall complex envelope of the transmitted signal $x(t)$ is

$$x(t) = \sum_i a_{I,i}s_m(t - iT) + j \sum_i a_{Q,i}s_n(t + \frac{T}{2} - iT) \quad (3)$$

where $s_m(t)$ and $s_n(t)$ have support only over $[-T/2, T/2]$. The received signal is passed through a noise limiting filter and sampled at the rate of r samples per symbol interval. Note that $x(t)$ is not a strictly bandlimited signal, and therefore, r must be large enough so that the Nyquist criterion is satisfied with respect to a modified bandwidth definition (say, -60 dB). The effects of r is investigated in [10], where it is found that the optimal receiver's performance is not degraded due to the sampling rate. However, suboptimal receivers may be seriously affected if a low sampling rate is employed.

Constrained Envelope Modulation (CEM): The CEM signal is generated by first obtaining a baseband equivalent linearly modulated signal $s_o(t)$ as

$$s_o(t) = \sum_{i=-N}^N a_i p(t - iT) \quad (4)$$

where $\{a_i\}$ is the sequence of M -ary symbols, $p(t)$ is a pulse-spreading filter, T is the symbol period, and $(2N + 1)$ is the total number of symbols transmitted. The transmit filter, in our study, is a root raised cosine (RRC) filter with a roll-off, α , $0 \leq \alpha \leq 1$. Therefore, the bandwidth of $p(t)$ is limited to $|f| \leq (1 + \alpha)/2T$. Thus, discrete time samples at the Nyquist rate and above can adequately describe the signal $s_o(t)$. In [7], a sampling rate $r = 2$ is considered. For a general integer r , the samples can be divided into r subchannels, so that the subchannel k , $0 \leq k \leq r - 1$, contains the samples $\{s_o(kT_r), s_o((k+r)T_r), s_o((k+2r)T_r), \dots\}$, where $T_r = T/r$. As shown in Fig. 1, all signal samples at the output of the symbol filtering unit pass through a delay unit on way to the combiner. The subchannel samples are, however, processed and appropriately filtered in order to constrain the amplitudes at the combiner output. Considering the subchannel 0, the i th sample is

$$s_o(iT) = \dots + a_{i-1}p(T) + a_i p(0) + a_{i+1}p(-T) + \dots \quad (5)$$

Since $p(0)$ is the largest magnitude of the filter (e.g., RRC) samples, the most significant contribution to $s_o(iT)$ comes from $a_i p(0)$. The remaining terms in (5) are individually small. However, depending on the symbols, they may all add up in correct phase to produce a large magnitude sample $s_o(iT)$ requiring larger transmitter power. To reduce this increased power requirement, a constrained generator is used as shown in Fig. 1. Each sample, $s_o(iT)$, from the output of filter $p(t)$ at point A is fed to the constrained-envelope generator for subchannel 0, and the magnitude of the sample, $|s_o(iT)|$, is compared to the threshold value $d_0 p(0)$, where d_0 is the largest magnitude of a signal point in the constellation diagram. If $|s_o(iT)| \leq d_0 p(0)$ then the discriminator output is made zero. However, if the signal magnitude $|s_o(iT)|$ exceeds the threshold $d_0 p(0)$, then the discriminator output generates an error signal sample $e_{0,i}$ as

$$e_{0,i} = s_o(iT) \left(\frac{|s_o(iT)| - d_0 p(0)}{|s_o(iT)| p(0)} \right) \quad (6)$$

This complex error sequence is passed through the bandlimited pulse-spreading filter $p(t)$ to generate the constrained bandwidth error signal for subchannel 0,

$$e^{(0)}(t) = \sum_i e_{0,i} p(t - iT) \quad (7)$$

This error signal is combined with $s_o(t)$, resulting in $s^{(0)}(t) = s_o(t) - e^{(0)}(t)$, so that the i -th sample is

$$s^{(0)}(iT) = s_o(iT) - s_o(iT) \left(\frac{|s_o(iT)| - d_0 p(0)}{|s_o(iT)|} \right) = s_o(iT) \left(\frac{d_0 p(0)}{|s_o(iT)|} \right) \quad (8)$$

Thus the magnitude of the combined signal sample $s^{(0)}(iT)$ is reduced to $d_0 p(0)$. Similarly, considering subchannel k , $k > 0$, the i -th sample at the output of the symbol filter is

$$s_o((k + ir)T_r) = \dots + a_{i-1}p((k + r)T_r) + a_i p(kT_r) + a_{i+1}p((k - r)T_r) + \dots \quad (9)$$

The error samples for subchannel k are then obtained as

$$e_{k,i} = s_o((k + ir)T_r) \left(\frac{|s_o((k + ir)T_r)| - d_0 p(0)}{|s_o((k + ir)T_r)| p(0)} \right) \quad (10)$$

This error sequence generates the constrained bandwidth error signal $e^{(k)}(t) = \sum_i e_{k,i} p(t - (k + ir)T_r)$. The combiner output, due to symbol filtering and subchannel processing, becomes a CEM signal

$$x(t) = s_o(t) - \sum_{k=0}^{r-1} e^{(k)}(t) = \sum_{i=-N}^N a_i p(t - iT) - \sum_{k=0}^{r-1} \sum_{i=-N}^N e_{k,i} p(t - (k + ir)T_r) \quad (11)$$

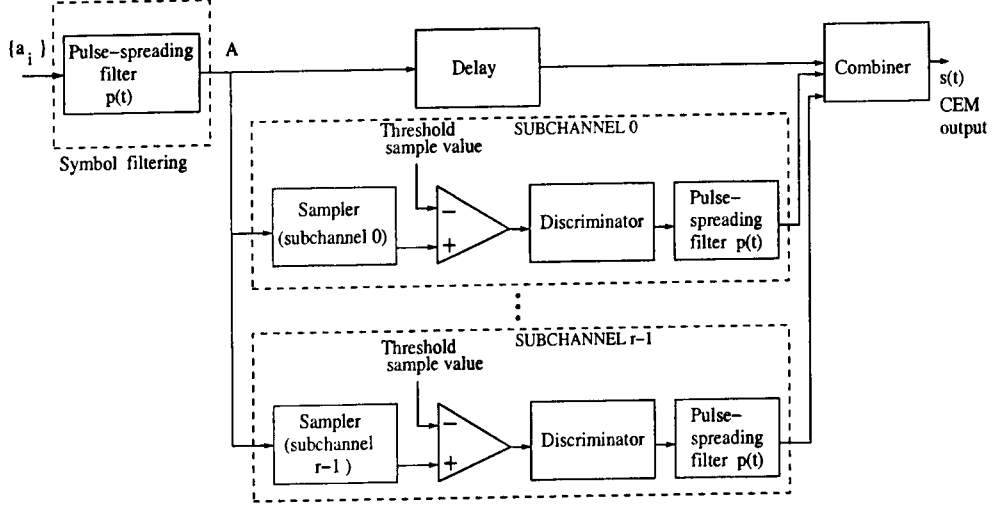


Fig. 1. Block diagram showing CEM generation

Note that the delay unit in the figure ensures correct timing at the combiner.

SPECTRAL PROPERTIES

Enhanced FQPSK: The EFQPSK signal can be described by

$$x(t) = \sum_i u(t - iT; a_i, \sigma_i) \quad (12)$$

where $\{a_i\}$ is the sequence of source symbols and $\{\sigma_i\}$ is a sequence of random variables, denoting the states of the modulator. In fact, EFQPSK is represented by a modulator with 16 states and four input symbols, denoted by the I and Q pair (0, 0), (0, 1), (1, 0), (1, 1). The waveforms $u(t; a_i, \sigma_i)$ take values from a set of deterministic, finite energy signals, comprising of $s_i(t)$, $0 \leq i \leq 7$, as described in (3). The power spectrum of such digital signals is [9]

$$G_x(f) = \frac{1}{T} \sum_{l=-\infty}^{\infty} [G_l(f) - G_{\infty}(f)] e^{-j2\pi f l T} + \frac{1}{T^2} G_{\infty}(f) \sum_{l=-\infty}^{\infty} \delta(f - \frac{l}{T}) \quad (13)$$

where $G_l(f) = E[U(f; a_{i+l}, \sigma_{i+l}) U^*(f; a_i, \sigma_i)]$, and $U(f; a_i, \sigma_i)$ is the Fourier transform of $u(t; a_i, \sigma_i)$. The term $G_{\infty}(f)$ is defined by

$$G_{\infty}(f) = \lim_{l \rightarrow \infty} G_l(f) = |E[U(f; a_i, \sigma_i)]|^2$$

The first term in (13) is line-free since the modulator memory is short and, therefore, $G_l(f) - G_{\infty}(f) \rightarrow 0$ rapidly as $l \rightarrow \infty$. This continuous part of the spectrum is studied in [1]. In this paper, our analysis focuses only on the line spectrum described by the second term in (13). The average spectrum $E[U(f; a_i, \sigma_k)]$ can be expressed as

$$E[U(f; a_i, \sigma_k)] = \sum_{k=0}^{15} \sum_{i=0}^3 P[\sigma_k] U(f; a_i, \sigma_k) P[a_i]$$

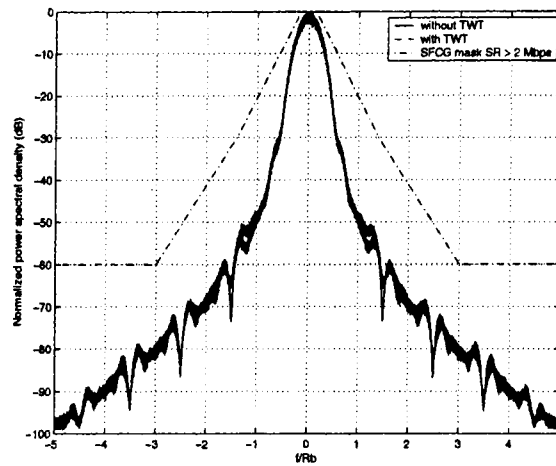


Fig. 2. Spectra of EFQPSK. The TWT operates at 0 dB backoff. The figure shows that the effects of the TWT on the spectrum are negligible.

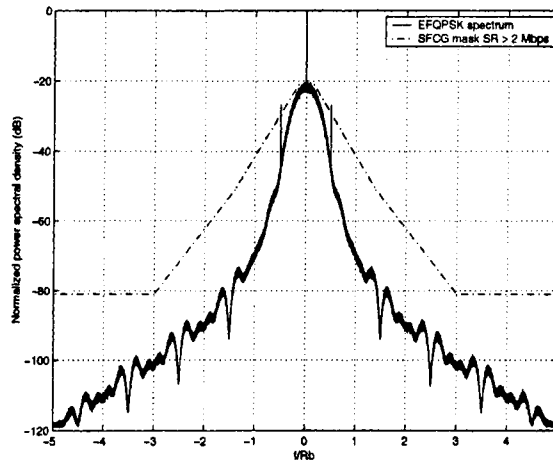


Fig. 3. Spectra of EFQPSK for unbalanced data (with +1 occurring 40% and -1 occurring 60%). The DC line is ignored in plotting the SFCG mask.

The spectrum $U(f; a_i, \sigma_k)$ for each state and each incoming symbol is obtained by taking the Fourier transform of the corresponding waveform. $P[\cdot]$ denotes probability.

Figures 2 and 3 show the simulated EFQPSK spectra for balanced and unbalanced data respectively. The space frequency co-ordination group (SFCG) spectral mask is also shown. The spectrum is observed to be well within the mask in the balanced data case. However, strong spectral lines are observed in the case of unbalanced data. The simulated strengths of the spectral lines are compared with the analytical results in Table 1. A normalization factor is assumed in the analytical results.

Constrained Envelope Modulation: The simulated spectra of CEM signals with QPSK modulation are shown in Figs. 4, 5, 6 and 7. Figure 4 shows that the spectral regrowth is not severe for CEM signals if the TWT model is an ideal clipper. This is a significant improvement over conventional QPSK without constrained envelope processing, and it agrees well with the observations in [6]. Figure 5 shows CEM spectra for a more realistic TWT model with both AM/AM and AM/PM effects, and

Probability of a bit being a +1	Relative strength at zero frequency (dB)		Relative strength at $f = 1/2T_b$ (dB)	
	Analysis	Simulation	Analysis	Simulation
0.45	+15.0	+16.0	-10.0	-10.0
0.40	+21.0	+21.0	-4.0	-4.0
0.35	+24.0	+25.0	-1.0	-0.5
0.30	+27.0	+30.0	+1.0	0.0
0.25	+29.0	+30.0	+2.0	+0.5
0.20	+30.0	+32.0	+2.0	+0.0

TABLE I

Spectral lines for unbalanced data. The relative strength measures strengths with respect to the spectrum level at zero frequency for balanced data.

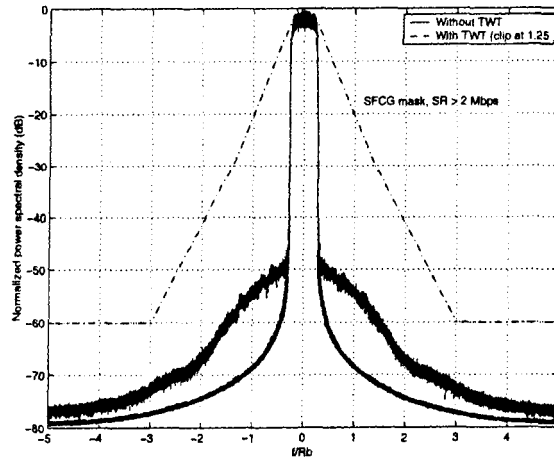


Fig. 4. Spectra of QPSK CEM with RRC filtering (roll-off factor=0.12). The TWT is an ideal clipper.

severe spectral regrowth is observed even with a 4 dB input backoff (IBO). Since CEM signal shows good spectral properties with an ideal clipper, a natural choice for improving its spectral properties is to use a predistorter. In this paper, the predistorter model described in [11] is used. The combination of the predistorter and the TWT produces an approximate clipper characteristics. The spectrum as a result becomes much more compact as shown in Fig. 6. Even for 0 dB input backoff, a comparison with Fig. 5 shows that the predistorter has improved upon the spectrum as shown in Fig. 7.

BIT ERROR RATE PERFORMANCE

Enhanced FQPSK: Different receiver structures for EFQPSK signals have been considered in [10]. The optimal receiver structure uses the Viterbi Algorithm (VA) with 16 states. Each state $(a_{I,k-1}, a_{I,k}, a_{Q,k-2}, a_{Q,k-1})$ represents 4 bits consisting of two I channel bits, $a_{I,k-1}, a_{I,k}$, and two Q channel bits, $a_{Q,k-2}, a_{Q,k-1}$. The transition from state $\sigma_l = (a_{I,k-2}, a_{I,k-1}, a_{Q,k-3}, a_{Q,k-2})$ to state $\sigma_p = (a_{I,k-1}, a_{I,k}, a_{Q,k-2}, a_{Q,k-1})$ is associated with the branch metric

$$\lambda(\sigma_l, \sigma_p, k) = \|y_{I,k} - a_{I,k}s_m\|^2 + \|y_{Q,k} - a_{Q,k}s_n\|^2$$

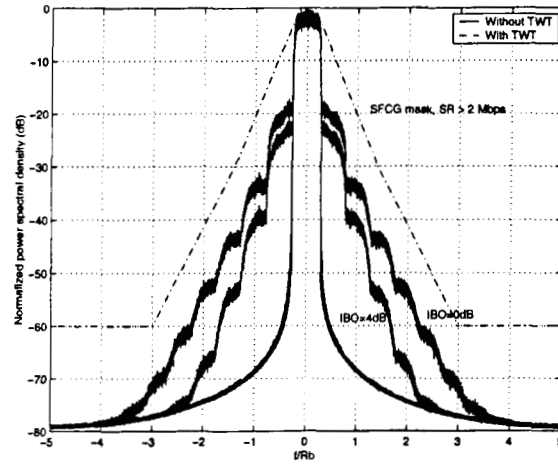


Fig. 5. Spectra of QPSK CEM with RRC filtering (roll-off factor=0.12) when a more realistic TWT model is used.

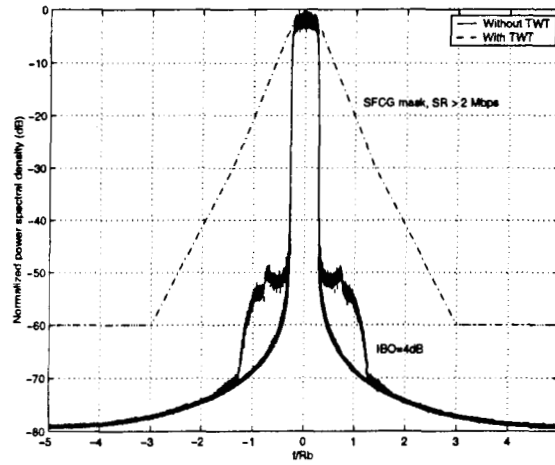


Fig. 6. Spectra of QPSK CEM with RRC filtering (roll-off factor=0.12). A more realistic TWT model and a predistorter are used.

where the waveform identification numbers m and n are obtained from (1) and (2) respectively, and $\|\cdot\|$ denotes Euclidean norm. The vector $\mathbf{y}_{I,k} = a_{I,k}\mathbf{s}_i + \mathbf{n}_I$, where the vector $\mathbf{s}_i$ consists of samples of $s_i(t)$, and $\mathbf{n}_I$ is the inphase noise sample vector. When a convolutional code of constraint length K is used, a joint detector and decoder implementation requires 2^{K+2} states. In this case, each state is associated with 6 encoder output channel bits. A state transition provides 2 additional bits. Therefore, each state transition provides 8 channel bits. These bits determine the waveform numbers (1) and (2), which are used in the calculation of the branch metric.

A suboptimal receiver structure can be realized using linear filters. Let $\mathbf{w}$ be a linear filter of r taps. Consider the detection of the I channel bit using this filter. Let the input signal vector to the filter while detecting the k -th bit be denoted by $\mathbf{y}_{I,k}$. This vector is of the form, $\mathbf{y}_{I,k} = a_{I,k}\mathbf{s}_i + \mathbf{n}_I$. The mean-squared error (MSE) at the output of the filter is $\mathcal{E} = E\{|\mathbf{w}^T(a_{I,k}\mathbf{s}_i + \mathbf{n}_I) - a_{I,k}|^2\}$. Taking the derivative of $\mathcal{E}$ with respect to $\mathbf{w}$ and setting it to zero, the optimal (Wiener) MMSE filter is

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\bar{\mathbf{s}} \quad (14)$$

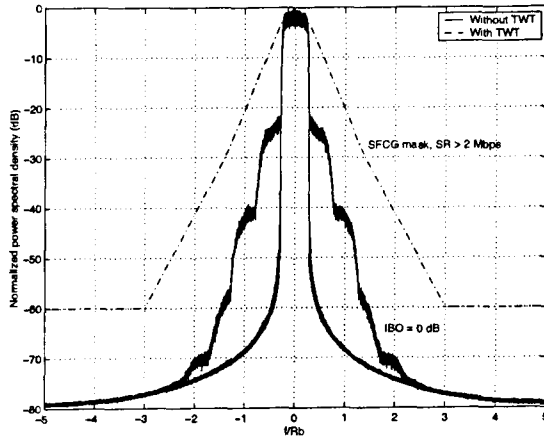


Fig. 7. Spectra of QPSK CEM with RRC filtering (roll-off factor=0.12). A more realistic TWT model and a predistorter are used.

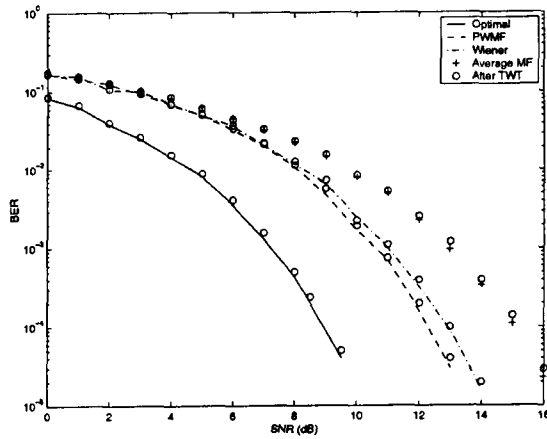


Fig. 8. BER performance of uncoded EFQPSK with and without a nonlinear amplifier.

where $\mathbf{R} = (1/8) \sum_{i=0}^7 \mathbf{s}_i \mathbf{s}_i^T + \sigma_n^2 \mathbf{I}$, $E\{\mathbf{n}_I \mathbf{n}_I^T\} = \sigma_n^2 \mathbf{I}$, and $\bar{\mathbf{s}} = (1/8) \sum_{i=0}^7 \mathbf{s}_i$. The same optimal filter is used for detecting both the I and Q channel symbols $a_{I,k}$ and $a_{Q,k}$ using the input vector $\mathbf{y}_{I,k}$ and $\mathbf{y}_{Q,k}$ respectively. Note that, at very low SNR, $\mathbf{R} \approx \sigma_n^2 \mathbf{I}$, and then, $\mathbf{w}_{opt} \approx \sigma_n^{-2} \bar{\mathbf{s}}$, which is the averaged matched filter receiver discussed in [2]. Note that the performance of the above linear filters can be slightly improved by using 8 parallel matched filters, each corresponding to one of the waveforms $\mathbf{s}_i$, $0 \leq i \leq 7$. This structure is called the Per-Waveform Matched Filter (PVMF) receiver [10].

The BER performance for uncoded EFQPSK is shown in Fig. 8. The performance degradation due to the TWT is found to be negligible in the SNR range shown. The TWT is operated with 0 dB IBO. The trellis coded Viterbi receiver's performance agrees with similar results reported in [2]. However, the average matched filter (MF) receiver's performance does not agree with the corresponding results in [2]. At high SNR, the performance gap among the receivers increases. Convolutional codes with $K = 7$, rate 1/2 with generators $g_1 = (1111001)$ and $g_2 = (1011011)$ are used in the study (Fig. 9). Coding is found to provide a gain of more than 4.5 dB at a BER of 10^{-3} for the optimal structure.

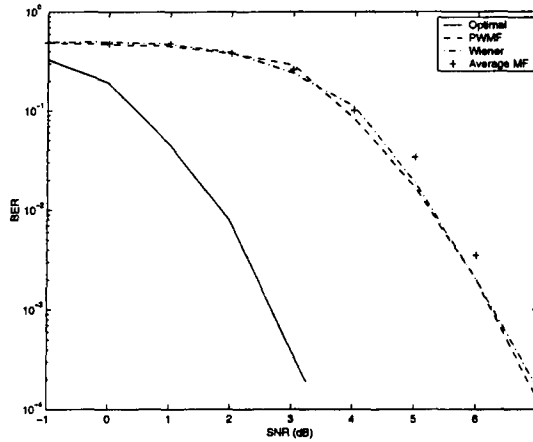


Fig. 9. BER performance of coded EFQPSK.

Constrained Envelope Modulation: The BER performance for uncoded and coded CEM signals is shown in Fig. 10. An MF receiver, a TWT at 0 dB IBO and a predistorter are used. The figure shows that the use of the TWT results in more than 1 dB loss. Note that a TWT actually amplifies an input signal and so the degradation in using the TWT should be seen as a loss in comparison with an ideal perfectly linear amplifier. Coding is found to provide more than 6 dB improvement at a BER of 10^{-4} .

A BER performance comparison of EFQPSK and CEM QPSK is shown in Fig. 11. The data rate and the TWT IBO (0 dB) are same for both the techniques. Thus the spectral efficiency of CEM is worse than that of EFQPSK in this case (Figs. 2 and 7). The EFQPSK uses the optimal receiver while the CEM uses an MF receiver. Although EFQPSK uses an optimal receiver, it can be implemented easily, and its complexity in the coded case is still not too high compared to the CEM/MF receiver. The figure shows that coded EFQPSK gains about 2 dB over coded CEM near a BER of 10^{-4} .

CONCLUSIONS

EFQPSK and CEM signals are studied. The advantages of EFQPSK include 1) Negligible degradation due to nonlinearities in a TWT. 2) Easily implementable optimal receiver structure, providing high power efficiency. The disadvantages of EFQPSK include 1) Sensitivity to unbalanced data, resulting in spectral lines. 2) Spectral efficiency with a linear amplifier is poorer than conventional QPSK. The advantages of CEM signals include 1) Insensitivity to unbalanced data, except a zero frequency DC line generation. 2) High spectral efficiency in conjunction with a predistorter. Moreover, higher order modulation can be used to improve spectral efficiency, albeit sacrificing power efficiency. The disadvantages of CEM include 1) Unknown power normalization factor. 2) Considerable spectral regrowth due to a TWT amplifier. Performance can degrade due to poor predistorter implementation.

REFERENCES

- [1] Simon M. K. and Yan T.-Y., "Unfiltered FQPSK: Another Interpretation and Further Enhancements, Part 1: Transmitter implementation and optimum reception", *Applied Microwave & Wireless*, pp. 76-96, Feb. 2000.
- [2] Simon M. K. and Yan T.-Y., "Unfiltered FQPSK: Another Interpretation and Further Enhancements, Part 2: Suboptimum reception and performance comparisons", *Applied Microwave & Wireless*, pp. 100-105, March 2000.

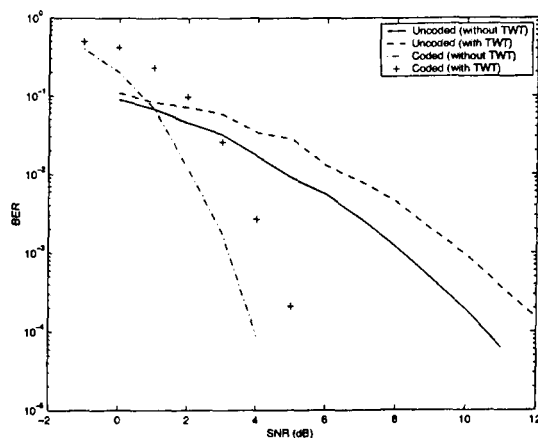


Fig. 10. BER performance of CEM QPSK

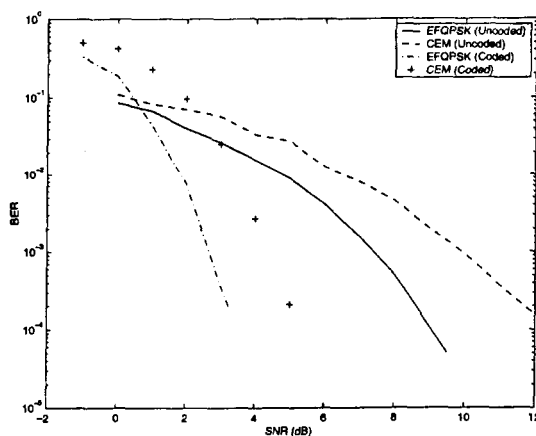


Fig. 11. BER performance comparison of EFQPSK and CEM QPSK

- [3] Kato S., and Feher K., "XPSK: A new cross-correlated phase shift keying modulation technique", *IEEE Trans. Commun.*, vol. 31, May 1983.
- [4] Molisch A. F., Editor, "Amplitude Limitation of OFDM Signals", *Wideband Wireless Digital Communications*, Prentice-Hall, Inc., New Jersey, 2001.
- [5] May T. and Rohling H., "Reducing the peak-to-average power ratio in OFDM radio transmission systems", Proc. VTC'98, Ottawa, Canada, 1998.
- [6] McCallister R., Putnam R., Andro M., and Fujikawa G., "Radiation-Hardened High-Performance Modulator Module", Proc. AIAA'00, Oakland, April 2000.
- [7] McCallister R. D., Cochran B. A. and Badke B. P., "Constrained-Envelope Digital-Communications Transmission System and Method Therefor", US Patent 6,104,761, Aug. 15, 2000.
- [8] Saleh A. A. M., "Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers", *IEEE Trans. Commun.*, COM-29, pp.1715-1720, Nov. 1981.
- [9] Benedetto S. and Biglieri E., *Principles of Digital Transmission with Wireless Applications*, Kluwer Academic/Plenum Publishers, New York, 1999.
- [10] Borah D. K. and Horan S., "Detection Techniques for Enhanced FQPSK Signals", Proc. IEEE Global Communications Conf., GLOBECOM 2001, San Antonio, Nov. 2001.
- [11] D'Andrea A. N., Lottici V. and Reggiannini R., "RF Power Amplifier Linearization Through Amplitude and Phase Predistortion" *IEEE Trans. Commun.*, vol.44, pp.1477-1484, Nov. 1996.